OAuth

OAuth is an authorization framework that enables applications to obtain limited access to a user's accounts and data. Using OAuth 2.0, Addepar can allow third-party applications to access a user's Addepar data using their Addepar credentials.

Authorization Code Grant Flow

This section illustrates how the Authorization Code grant process works.

1. Your end-user will access the Addepar integration through your client application.

2. Your client application will send the user to the Addepar login modal using the authorization request URL, which includes your redirect URI.

3. Upon login, the user will be prompted to authorize Addepar scopes.

4. Once the user has authorized access to your client application, they will be sent back to the redirect URI from step 2, along with a unique authorization code in the URL.

5. The client application takes the authorization code it receives from Addepar and makes another request to the server. This request includes the OAuth client secret key. When the server sees a valid authorization code and client secret key, it responds with an access token that is valid for 60 minutes, and a refresh token that must be stored within your client application for subsequent requests.

6. At this point, the user is free to make requests to Addepar through your client application until their access token expires, which requires the stored refresh token and secret to request a new access token.

OAuth Client Setup

This section outlines what Addepar provides you and what you need to provide to get set up as an OAuth client.

Addepar provides the following:

- Client ID: a string. For example, this could be a firm's name.
- Client Secret: a 256-bit value converted to a hexadecimal string representation.
- The list of scopes that the client application can access once it's authorized.

The Client ID and Client Secret are used when the client application exchanges the authorization code for access and refresh tokens.

Provider of the client application must supply the following:

- Redirect URI: A URL within the client application that the OAuth server will redirect to upon a successful authorization. The client application should implement server-side logic to complete the OAuth handshake, i.e., an authorization code exchange for access and refresh tokens. Multiple redirect URLs are supported.
- Terms of Service URL.
- Privacy Terms URL.

Scopes

SCOPES	READ OR WRITE
PROFILE	Read only access to your name, email address, unique user ID, and Addepar firm ID. This information will be used to identify your Addepar account.
PORTFOLIO	Read only access to each client's portfolio data and any attributes, including but not limited to name, quantity, and value of all clients, legal entities, accounts, and securities.
TRANSACTIONS or TRANSACTIONS_WRITE	Read-only or read and write access to each client's transactions data, including but not limited to name, direct owner, transaction type, and value of transactions.
FILES or FILES_WRITE	Read-only or read and write access to the name and content of all files associated with each client. Files may contain portfolio data and any attributes, including but not limited to name, quantity, and value of all clients, legal entities, accounts, and securities.
GROUPS or GROUPS_WRITE	Read-only or read and write access to each group's details and any attributes applied to those groups, including but not limited to name, date created, and a list of clients, accounts, and entities in the group.
ENTITIES or ENTITIES_WRITE	Read-only or read and write access to entities and any associated attributes, including but not limited to clients, trusts, holding accounts, and investments.
POSITIONS or POSITIONS_WRITE	Read-only or read and write access to positions that define ownership between two entities and any attributes applied to those positions.
USERS or USERS_WRITE	Read-only or read and write access to users' details in your firm, including but not limited to name, email, roles, and any associated attributes.
AUDIT_TRAIL	Read-only access to audit logs and any associated attributes, including but not limited to transactions, reports, user roles, and permissions.

Request User Authorization

The example below shows a sample authorization URL. Your Addepar contact will provide you with the correct authorization URL to use for your integration or OAuth client.

Send Authorization Request: From a Developer's Perspective:

1. You'll start by constructing a URL as detailed in the table below. This will redirect the user to the URL to begin the authorization process.

≡The redirect URI must be URL encoded.

2. Once the user has authorized the integration, they will be redirected back to the provided redirect URI along with the authorization code appended to the end.

3. You'll then fetch the authorization code from the incoming request URL and make the API request to gain access and refresh tokens with the provided authorization code.

≡The authorization code expires after 60 seconds. After 60 seconds the user will have to reauthorize against Addepar to generate another authorization code.

Send Authorization Request: From a User's Perspective:

1. The user starts the authorization process from a third party application where they will be redirected back to Addepar to login and authorize the integration.

2. Once the user logs in, they will select the appropriate firm. If they have access to only one firm, they will click Continue. If they have access to multiple firms, they will click Continue for the firm they want to set up the integration for.

3. They will then click Authorize to allow third-party integration to gain access to the appropriate scope(s).

4. After authorizing, the user will be redirected back to the third-party application along with the authorization code.

Method	**GET**
URL	Example: https://id.addepar.com/oauth2/authorize? response_type=code&scope=portfolio&client_id=example&state=state&redirect_uri=https%3A%F%2Fexample.com%2Foauth%2Fcb *This is an example URL. Reach out to your Addepar contact for your unique URL.*

Request Access or Refresh Tokens

≡You can only make this request once per authorization code. Save the resulting refresh token in a secure location. The access_token is good for 60 minutes and can be renewed indefinitely with the refresh token.

Request Access

≡To set up your OAuth client, reach out to your Addepar contact for assistance.

Using Postman or cURL, prepare an API request, as shown below.

Method	**POST**
URL	Production: https://api.addepar.com/public/oauth2/token Development Environment: https://examplefirm.clientdev.addepar.com/api/public/oauth2/token
Form Parameters	client_id: example client_secret: VQ2b71RqtYloj//enHEdALepvo5s7Huobbboo9KQMj8= grant_type: authorization_code redirect_uri: https://example.com/oauth/cb code: CAEQBRoQUHOKhNgeqAtSXCwkh54eYg

Example:

```
REQUEST     RESPONSE

curl --location --request POST 'https://examplefirm.addepar.com/api/public/oauth2/token' \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'client_id=example' \
  --data-urlencode 'grant_type=authorization_code' \
  --data-urlencode 'redirect_uri=https://example.com/oauth/c' \
  --data-urlencode 'code=CAEQBRoQUHOKhNgeqAtSXCwkh54eYg' \
  --data-urlencode 'client_secret=VQ2b71RqtYloj//enHEdALepvo5s7Huobbboo9KQMj8='
```

Response Codes

- 200 OK : Success
- 400 Bad Request : One of the request parameters was invalid or the authorization code was expired

Refresh Token

Using Postman or cURL, prepare an API request as shown below.

Method	**POST**
URL	Production: https://api.addepar.com/public/oauth2/token Development Environment: https://examplefirm.clientdev.addepar.com/api/public/oauth2/token
Form Parameters	client_id: example client_secret: VQ2b71RqtYloj//enHEdALepvo5s7Huobbboo9KQMj8= grant_type: refresh_token refresh_token: CAEQAxogfUWVFDflDMBqSYVWp4m7A4-yLwUoCZ0svXwsrl7hKN8

Example:

REQUEST   RESPONSE

```
curl --location --request POST 'https://examplefirm.addepar.com/api/public/oauth2/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=example' \
--data-urlencode 'grant_type=refresh_token \
--data-urlencode 'refresh_token=VONGFSZ4SG5R3KNZVYHT926I71VVTQ' \
--data-urlencode 'client_secret=VQ2b71RqtYloj//enHEdALepvo5s7Huobbboo9KQMj8='
```

Response Codes

- 200 OK : Success
- 400 Bad Request : One of the request parameters was invalid or the authorization code was expired

Example: Make a Request

After you've gained an access token, you can begin sending requests to the Addepar APIs. The Addepar API documentation provides all available details on each API endpoint.

Using a portfolio query as an example, prepare an API request using Postman or cURL, as shown below.

Method	**POST**
URL	Production: https://api.addepar.com/v1/portfolio/query Development Environment: https://examplefirm.clientdev.addepar.com/api/v1/portfolio/query
Headers	Authorization: Bearer CAEQAhoQABpusROJW_ABoeI9eW9ZjA Content-Type: application/vnd.api+json Addepar-Firm: 1

Example:

```
REQUEST    RESPONSE

curl -X POST \
  https://api.addepar.com/api/v1/portfolio/query \
  -H 'Content-Type: application/vnd.api+json' \
  -H 'Addepar-Firm: `1' \
  -H 'Authorization: Bearer CAEQAhoQG4WaecpqShCPSOzjGt6u-A' \
  -d \
'{
  "data": {
    "type": "portfolio_query",
    "attributes": {
      "columns": ["value"],
      "groupings": [],
      "start_date": "2018-01-01",
      "end_date": "2019-01-01",
      "portfolio_type": "FIRM",
      "portfolio_id": 1
    }
  }
}'
```

Response Codes:

- `200 OK` : Success
- `400 Bad Request` : Improperly formatted query or lacking necessary data permissions
- `403 Forbidden` : User lacks API permission or has not granted the appropriate scope

What's Next

Response Codes

ADDEPAR

Addepar is a wealth management platform
that specializes in data aggregation,
analytics and performance reporting.

OUR APIS

Attributes API

Entities API

Entity Types API

Files API

Groups API

Jobs API

Portfolio API

Positions API

DEVELOPERS

Documentation

API Explorer

MORE

Addepar Home

Engineering Blog

Integrations

Contact Us

Partner with Addepar

Roles API

Transactions API

Users API

Privacy Policy | Terms of Use